



BOOM
AiGeneration

AGENTI GPT

COSA SONO E COME FUNZIONANO

GUIDA COMPLETA: DEEP DIVE 5

LA FUNZIONE PROGRAMMING

SWIPE FOR MORE



Gli agenti GPT con funzionalità **"Programming"** sono strumenti avanzati progettati per assistere gli sviluppatori nella scrittura, ottimizzazione e debugging del codice.



Questi agenti possono **generare codice** in diversi linguaggi, **suggerire miglioramenti**, **risolvere bug** e **spiegare concetti complessi** legati alla programmazione, rendendo lo sviluppo software più efficiente e accessibile.



Un esempio pratico di tale tecnologia è **Python Assistant AI**, un agente specializzato nel supporto alla programmazione Python, con capacità di refactoring, test automation e analisi delle performance.

Esempio pratico

Automazione della gestione dei file in una cartella condivisa

Un'azienda ha bisogno di un sistema automatizzato per **monitorare una cartella condivisa** e **organizzare i file in base alla tipologia** (es. immagini, documenti, fogli di calcolo).



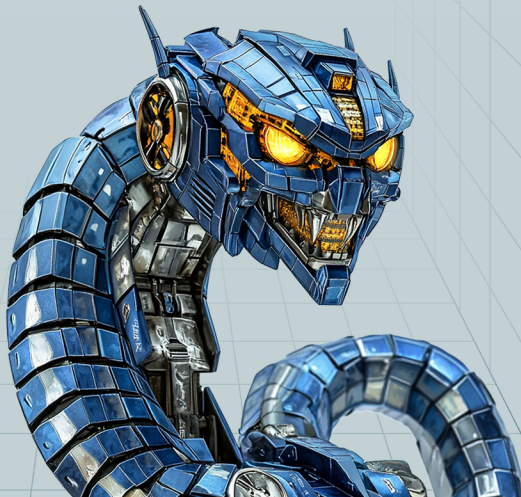
L'agente GPT genera uno **script Python che:**

1. Monitora continuamente la cartella:
l'agente utilizza watchdog per il monitoraggio in tempo reale della cartella WATCH_FOLDER e shutil per lo spostamento dei file

2. Classifica i file in base all'estensione

3. Sposta i file nelle relative sottocartelle

4. Rinomina i file seguendo una convenzione (es. data_tipo_nomefile)



Di seguito il codice che automatizza le singole attività:

```
import os
import shutil
import time
from datetime import datetime

# Percorso della cartella da monitorare
cartella_monitorata = "./cartella_condivisa"

# Sottocartelle per le diverse tipologie di file
tipi_file = {
    "Immagini": [".jpg", ".png", ".gif"],
    "Documenti": [".pdf", ".docx", ".txt"],
    "Fogli_Calcolo": [".xlsx", ".csv"]
}

# Funzione per classificare e spostare i file
def organizza_file():
    for file in os.listdir(cartella_monitorata):
        percorso_completo = os.path.join(cartella_monitorata, file)
        if os.path.isfile(percorso_completo):
            estensione = os.path.splitext(file)[1].lower()
            for categoria, estensioni in tipi_file.items():
                if estensione in estensioni:
                    nuova_cartella = os.path.join(cartella_monitorata, categoria)
                    os.makedirs(nuova_cartella, exist_ok=True)
                    nuovo_nome = f"{datetime.now().strftime('%Y%m%d_%H%M%S')}_{categoria}_{file}"
                    shutil.move(percorso_completo, os.path.join(nuova_cartella, nuovo_nome))
                    print(f"Spostato: {file} → {categoria}/{nuovo_nome}")

# Monitoraggio continuo della cartella
if __name__ == "__main__":
    while True:
        organizza_file()
        time.sleep(5) # Controllo ogni 5 secondi
```



Alcune migliorie utili che potrebbero essere fatte allo script:

- **Gestione degli errori migliorata:** aggiungere un blocco “try-except” per evitare crash se un file è in uso o se ci sono problemi di permessi
- **Supporto per file compressi:** aggiungere una categoria per “.zip”, “.rar”, “.tar.gz”, ecc., per estrarli automaticamente
- **Registro delle operazioni (logging):** salvare i dettagli dei file spostati in un file di log per tracciare l'attività dello script



ESEMPIO DI LOGGING

```
import logging
logging.basicConfig(filename='file_organizer.log', level=logging.INFO)
```



Altre funzionalità di Python

- **Modificare il codice** o aggiungere una parte di codice in base al feedback dell'agente.



```
24  class Test():  
25      prit("ciao")
```

why doesn't work?

There's a typo in your code:

python

 Copy

 Edit

```
class Test():  
    prit("ciao")
```

Issues:

1. `prit` is not a valid function – It should be `print("ciao")` .
2. `Test` is an empty class – If you intended to define a class, you need a constructor (`__init__`) or methods.

Fix:

python

 Copy

 Edit

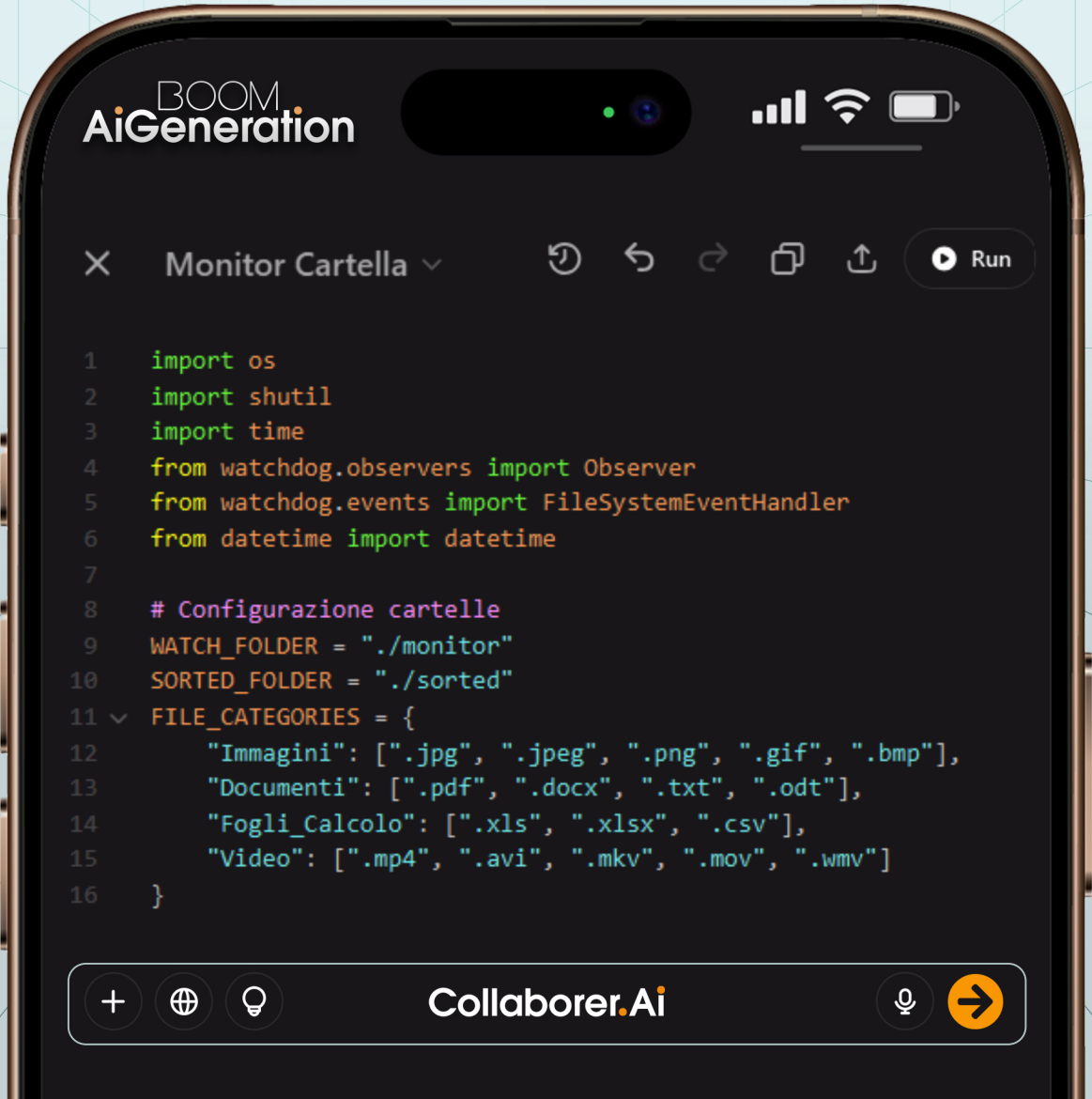
```
class Test:  
    def __init__(self):  
        print("ciao")
```

Would you like me to apply this fix in your document? 



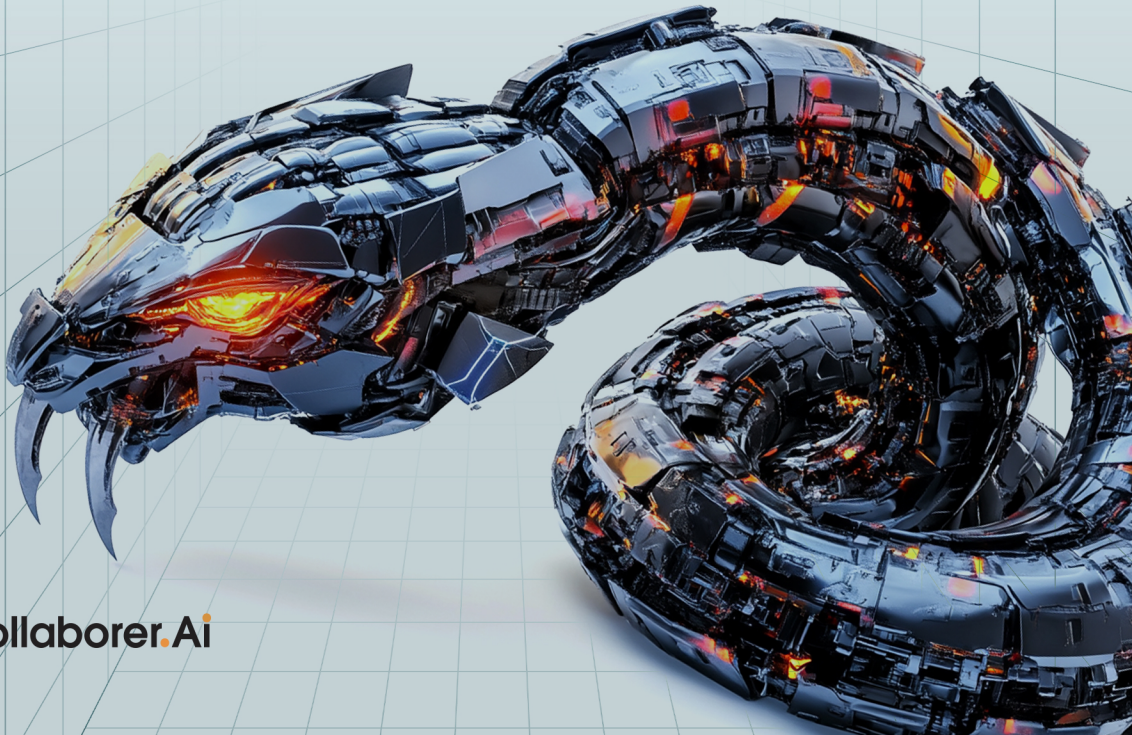
Altre funzionalità di Python

- Eseguire il run del codice



In conclusione

Gli agenti GPT con funzione **Python** semplificano l'automazione di attività ripetitive come la gestione di file. In pochi secondi, generano codice ottimizzato, suggeriscono miglioramenti e creano test automatizzati, **migliorando efficienza e affidabilità.**



E tu?

La tua organizzazione ha già sperimentato l'automazione di task ripetitivi tramite agenti GPT? Quali risultati avete ottenuto?

Scrivilo nei **commenti!**

Seguici e unisciti a noi
per diventare parte
della Community

BOOM
AiGeneration
Collaborer.Ai

